AFOSR-TR-97-0646

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
April 29, 1997

3. REPORT TYPE AND DATES COVERED
Final Technical Report 1 Nov 93 to 31 Jan 97

4. TITLE AND SUBTITLE
Knowledge-Based Decision Model Construction for Dynamic Interpretation Tasks

5. FUNDING NUMBERS
F49620-94-1-0027

6. AUTHOR(S)
Michael P. Wellman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Michigan
Artificial Intelligence Laboratory
1101 Beal Avenue
Ann Arbor, MI 48109-2110

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
110 Duncan Avenue, Ste B115
Bolling AFB, DC 20332-8050

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release, Distribution Unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The aim of this project was to identify general principles and develop concrete techniques for knowledge-based construction of probabilistic models supporting dynamics decision making under uncertainty. We focused on problems where the precise decision context (i.e., which options are available and what information is known) is highly variable, precluding specification of a fixed model in advance. The project yielded technical results in four areas of reasoning and decision making under uncertainty involving model construction: (1) path planning and scheduling under uncertainty, (2) abstraction and other approximation methods for Bayesian networks, (3) Bayesian methods for pattern and plan recognition, and (4) aggregation of beliefs across multiple agents.

14. SUBJECT TERMS

15. NUMBER OF PAGES
20

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFOSR Final Technical Report

Covering the period 1 Nov 93 to 31 Jan 97

Project: **Knowledge-Based Decision Model Construction for Dynamic Interpretation Tasks**

Grant Number: **F49620-941-0027**

Principal Investigator: **Michael P. Wellman**

University of Michigan
Artificial Intelligence Laboratory
1101 Beal Avenue
Ann Arbor, MI 48109-2110

Abstract

The aim of this project was to identify general principles and develop concrete techniques for knowledge-based construction of probabilistic models supporting dynamic decision making under uncertainty. We focused on problems where the precise decision context (i.e., which options are available and what information is known) is highly variable, precluding specification of a fixed model in advance. The project yielded technical results in four areas of reasoning and decision making under uncertainty involving model construction: (1) path planning and scheduling under uncertainty, (2) abstraction and other approximation methods for Bayesian networks, (3) Bayesian methods for pattern and plan recognition, and (4) aggregation of beliefs across multiple agents.

1. Overview

The aim of this project was to identify general principles and develop concrete techniques for knowledge-based construction of probabilistic models supporting dynamic decision making under uncertainty. We focused on problems where the precise decision context (i.e., which options are available and what information is known) is highly variable, precluding specification of a fixed model in advance.

Many distinct problems of reasoning and decision making under uncertainty involve a substantial model construction component,[1] and we explored several of these. The project yielded technical contributions in four specific areas: (1) path planning and scheduling under uncertainty, (2) abstraction and other approximation methods for Bayesian networks, (3) Bayesian methods for pattern and plan recognition, and (4) aggregation of beliefs across multiple agents. We describe the results of our investigations in each of these areas in the sections below.

Our work has combined algorithm design, theoretical analysis, and empirical evaluation. All of our algorithms have been implemented, and our growing body of code provides the basis for further research in these areas.

2. Path Planning and Scheduling

One class of problems in uncertain reasoning explored in this project is that of path planning and scheduling under uncertainty. For cases where the operator costs are deterministic and solely a function of the current state, the well-known A* state-space search technique guarantees optimal paths. However, when costs are stochastic and path dependent, A* may prune partial paths that could lead to superior solutions.

Path-dependent costs occur in situations where the cost, c_{ij}, of applying operator j in state i depends on the cost of the path taken to that state. One source of path dependence is a utility function that is nonlinear in cost, for example if cost is measured in time and utility is based on meeting a deadline.

It has been shown (Kaufman and Smith 1993) that A* produces optimal

[1] Evidence of this is growing recognition of the model construction task within the Uncertain Reasoning research community. The principal investigator organized a workshop on Knowledge-Based Model Construction of Probabilistic and Decision Models at AAAI-91. With the co-organizers, he edited a special issue of the journal *IEEE Transactions on Systems, Man, and Cybernetics* on this topic, which appeared in November 1994. Numerous subsequent articles have been published by other researchers on the topic, and there was a panel discussion by this name at the Conference on Uncertainty in AI, 1996.

solutions even with path-dependent cost functions, as long as a particular *consistency*, or *monotonicity*, condition applies. The monotonicity condition demands that for any path costs $c \leq c'$,

$$c + c_{ij}(c) \leq c' + c_{ij}(c'). \qquad \text{(Eq. 1)}$$

Note that this form of monotonicity—on the accumulated path cost—is weaker than requiring monotonicity of c_{ij} itself.

It follows from this condition that for two paths, A and A', both leading to the same state, the superiority of one, $Cost(A) \leq Cost(A')$, implies that the same relation holds for these paths extended by a given action, a, that is, $Cost(Aa) \leq Cost(A'a)$. Given this result, it is safe to prune A' because for any path to the goal based on that path, there is a path at least as good based on A.

A second variation of standard state-space search is to admit stochastic costs, that is, to treat c_{ij} as a random variable. If c_{ij} depends only on the state, and utility is linear in cost (i.e., the agent is risk neutral), then it is sufficient to use A* with operator costs represented by their means.

However, if the problem requires both stochastic and path-dependent operator costs, then we are no longer justified in pruning paths based upon expected costs. Rather we must use the Stochastic Dominance A* (SDA*) algorithm (Wellman et al. 1995). There are four primary differences between SDA* and A*.

Stochastic Monotonicity: We require a stochastic version of the monotonicity condition used to address path dependence in the deterministic case. Specifically, for all costs c, c', and z, $c \leq c'$ implies

$$\Pr(c + c_{ij}(c) \leq z) \geq \Pr(c' + c_{ij}(c') \leq z). \qquad \text{(Eq. 2)}$$

Pruning: Rather than keeping the single lowest-cost path to a node, we must keep all of the *admissible* paths, where admissibility is defined by *stochastic dominance*. We have shown that if paths A and A' lead to the same state and A stochastically dominates A', $Cost(A) \leq_{SD} Cost(A')$, then A' cannot be part of a uniquely optimal solution. Specifically, the stochastic monotonicity condition in this situation entails that for any incremental action a, $Cost(Aa) \leq_{SD} Cost(A'a)$. If, however, A' is not stochastically dominated, then it is possible to construct an example where it does in fact lead to the optimal solution.

Heuristics: When costs are path dependent, the heuristic functions must likewise take into account path cost as well as the state. And whereas a heuristic is admissible for A* if it underestimates cost to the goal, for the stochastic case an admissible heuristic must produce estimated cost *distributions* that *stochastically dominate* the actual cost distribution.

Priority: Search nodes are expanded in order of estimated expected utility. Like A*, the algorithm terminates when a goal node is popped off the priority queue. The reasoning is as follows: given that the heuristic function is stochastically admissible, and the accumulated path costs stochastically monotone, expected utility is monotonically decreasing along a path. Thus, when a solution is found, any intermediate path that had an estimated expected utility less than that of the solution must have already been explored or pruned.

Under the conditions described above, SDA* provides an optimal and complete solution procedure for problems with path-dependent stochastic operator costs. In Table 1a, we see that path-dependence alone can be accommodated by a monotonicity condition (Kaufman and Smith 1993), and stochastic costs alone by using means, but the conjunction of both requires SDA* (Wellman, Ford et al. 1995).

	State Dependent	Path Dependent
Deterministic Costs	A*	A*
Stochastic Costs	A* with means	SDA*

(a) scalar costs

	State Dependent	Path Dependent
Deterministic Costs	MOA*	MOA*
Stochastic Costs	MOA* with means	MO-SDA*

(b) multidimensional costs

Table 1: Appropriate search methods for combinations of three cost types. All path-dependent cases require a (deterministic or stochastic) monotonicity condition.

Stochastic-dominance pruning can yield enormous savings over exhaustive search. On a representative series of path planning problems on *n*-by-*n* grids, we performed a simple test of the algorithm. As illustrated in Figure 1, exhaustive search is intractable for relatively small grid sizes, whereas search with dominance pruning enables us to solve substantially larger problems, with no sacrifice in solution quality.

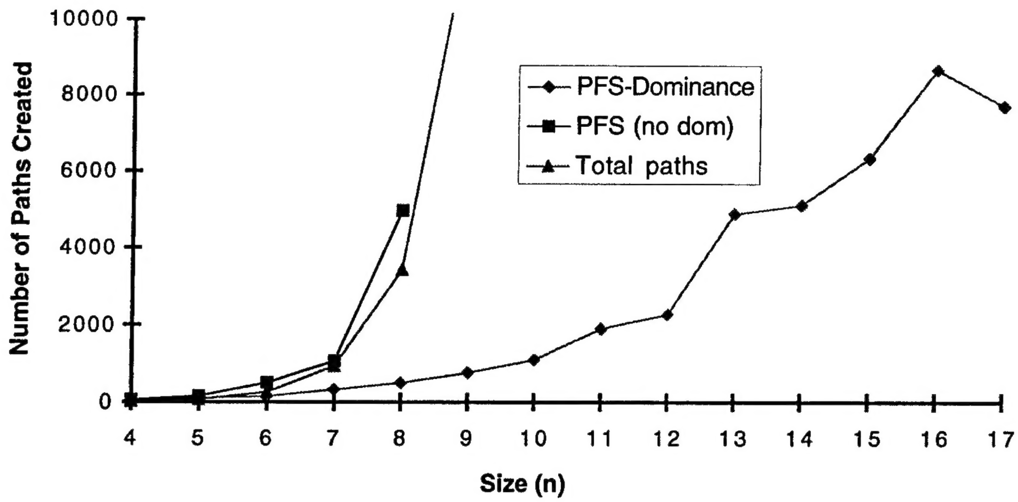

Figure 1: Number of paths created as a function of grid size, with and without dominance pruning.

However, in our effort to apply SDA* to a problem of factory scheduling under uncertainty (Wurman and Wellman 1996), we required a further extension. The utility function employed in this problem depends on the tardiness penalty of *each* order, and thus our cost is not expressible as a scalar quantity (without undue expansion of the state space). We therefore define a two-dimensional cost structure, representing the penalty accumulated for orders completed thus far, along with the time taken along the path. Fortunately, the extension of A* to multidimensional costs has already been investigated by Stewart and White (Stewart and White 1991), who proposed the Multiobjective A* (MOA*) algorithm for this case.

Like SDA*, MOA* extends A* by pruning paths based on dominance rather than point utility. With multidimensional costs, it is generally not possible to order intermediate solutions. Under the assumption that utility is nonincreasing in each cost dimension, however, the policy of pruning based on dominance guarantees finding all optimal solutions. This result can be extended to stochastic and path-dependent costs in a manner analogous to the scalar case, as we diagram in Table 1b. The particular contribution of our work on factory scheduling under uncertainty (Wurman and Wellman 1996) lies in the lower right cell of this table.

Our factory scheduling algorithm is the first general search procedure that exploits stochastic dominance to produce optimal schedules under time-dependent uncertainty. Although uncertainty is ubiquitous in factory scheduling problems (e.g., uncertain setup and production times), previous research has tended to ignore this topic, or give up on exact algorithms.

3. Abstraction and Approximation

3.1 State-Space Abstraction

One part of this project was devoted to investigating the idea of approximating Bayesian networks by selectively ignoring distinctions in the state space of random variables (Wellman and Liu 1994). By solving the network using progressively refined state variables, we can achieve whatever level of accuracy time will allow, while still providing useful results even in highly time-stressed situations.

Our anytime approximation algorithm works as follows. Given the original Bayesian network (OBN), we construct and evaluate an abstract Bayesian network (ABN) in each iteration. We start with the most extreme abstraction, where the state spaces of all non-evidence nodes are collapsed into a single aggregate state, or *superstate*. We then evaluate the network to obtain an approximation of the desired probability distribution, along with information regarding how we may refine the superstates in subsequent iterations. If there is time available, we then select and split a superstate in an abstracted node, resulting in a new, more complex ABN. The algorithm continues in this way until either we have no way or no need to refine the state space, or there is no time available for more computation. A description of the algorithm is presented in Figure 2.

procedure Abstract-Iter(OBN, evidence)

1. Generate an initial ABN with one superstate per abstracted node.
2. Evaluate the probability distribution for each node given the evidence.
3. If all states for all nodes are elementary, return.
4. Split a superstate in an abstracted node.
5. Go to step 2.

Figure 2: The iterative abstraction procedure.

This approach is similar to others being studied, and therefore we have framed our theoretical results in a general enough context to cover many forms of abstraction and approximation. In brief, our basic result shows how to compare the quality of abstraction approximations based on the position of a node in the network structure (Liu and Wellman 1995). These sorts of results are potentially useful because any approximation algorithm has to choose how to allocate its computational effort for any uncertain reasoning task.

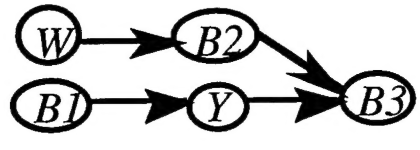

Figure 3: A simple Bayesian network.

To measure the quality of approximations, we apply the *Kullback score, K,* which increases with the divergence between the true and the approximated distributions. We have shown that the quality of the approximated distribution of a set of variables that d-separates another set of variables from the abstracted variables is not better than that of the d-separated set. Consider the Bayesian network shown in Figure 3 above. If we instantiate variable *B3* and abstract *B1*, then the qualities of the approximated marginal distributions of other variables have the relationship: $K_Y \geq K_{B2} \geq K_W$.

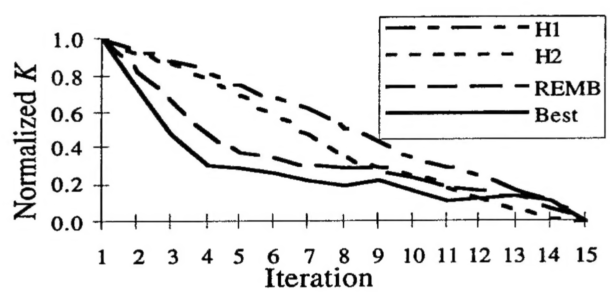

Figure 4: Comparison of heuristics for controlling state-space abstraction.

We are searching for principled methods to control the state-space abstraction process, aiming at improving performance profiles of the algorithm. In particular, we are exploring score functions for selecting the aggregated states to refine next. We aim for functions that are highly correlated with the actual improvement of the approximation, yet based on factors that are local to the aggregated states. Among the functions being investigated, the REMB score, based on the relative entropy of the Markov boundary of the abstracted variable, has demonstrated the best average performance in experiments. This is illustrated by the closeness of the REMB curve to the "Best" curve in the performance graph presented in Figure 4. "Best" represents the actually optimal choice, based on complete evaluation. The H1 and H2 curves in the figure represent other intuitive heuristics based on probability mass and skewness.

Continuing work has continued our theoretical and empirical analysis of abstraction approximations, and in particular examines the usage of *value of information* in control regimes.

3.2 Incremental Tradeoff Resolution

In work prior to this project, the investigator developed a formalism for qualitative probabilistic reasoning, based on specifications of the signs of influence among nodes in a Bayesian network (Wellman 1990). Use of qualitative models was motivated primarily on knowledge representation grounds, supporting derivation of properties of decisions without full numeric specification of the probabilistic model.

However, it may also be possible to exploit qualitative information to improve the efficiency of probabilistic inference. Specifically, we consider the task of deriving the qualitative relationship (i.e., the sign of the probabilistic association) between a pair of variables in a Bayesian network. From an abstracted version of the network, where all local relationships are described qualitatively, we can derive the entailed sign between the variables of interest efficiently using propagation techniques (Druzdzel and Henrion 1993). However, since the abstraction process discards information, the result may be qualitatively ambiguous even if the actual relationship entailed by the precise model is not.

Nevertheless, there are several ways to exploit qualitative reasoning to derive these relationships without necessarily resorting to solution of the complete problem at full precision, even in cases where purely qualitative reasoning would be ambiguous. We have investigated incremental approaches, in which we apply numeric reasoning to either subproblems or simplified versions of the original, to produce an intermediate model more likely to be qualitatively unambiguous (Liu and Wellman 1997).

For example, consider the qualitative probabilistic network (QPN) shown on the left-hand side of Figure 5. Since there exist both a positive path (through X) and a negative path (direct arc) from W to Z, the qualitative influence of W on Z is ambiguous. This local "?" would propagate throughout the network, necessarily ambiguating the relationship of any predecessor of W to any successor of Z.

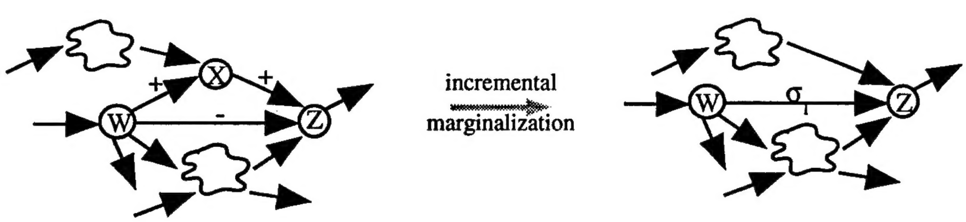

Figure 5: Marginalizing X potentially resolves the qualitative influence of W on Z.

Once we have detected the source of such a local ambiguity, we may attempt to resolve it by marginalizing node X. The new sign on the direct arc from W

to Z can be determined by inspecting the new conditional probability table of Z. If we are fortunate, the qualitative sign will turn out to be decisive, in which case we have resolved the tradeoff.

This example illustrates the main idea of the incremental marginalization approach to resolving tradeoffs in QPNs. If we get an unambiguous answer from the reduced network after marginalizing a selected node, then there is no need to do further computation. If the answer is still ambiguous, we may select other nodes to marginalize. The iteration continues until a decisive answer is uncovered. We present the skeleton of the Incremental TradeOff Resolution algorithm below. The algorithm is designed to answer queries about the qualitative influence of a *decision* node on some *target* node, using some *strategy* for selecting the next node to reduce.

ITOR(*decision,target,strategy*)

1. Remove nodes that are irrelevant to the query about *decision*'s influence on *target* (Shachter 1988).

2. Attempt to answer the query via qualitative inference (Druzdzel and Henrion 1993).

3. If the answer to the query is decisive, exit; otherwise continue.

4. Select a node to reduce according to *strategy*, perform the node reduction, and calculate the qualitative abstractions of the transformed relationships. Return to Step 2.

In general, we expect the incremental approach to improve performance over purely numeric inference. Since qualitative inference is quadratic whereas exact inference in Bayesian networks is exponential in the worst case, the qualitative inference steps do not add appreciably to computation time. On the other hand, when the intermediate results suffice to resolve the tradeoff, we save numeric computation over whatever part of the network is remaining.

4. Pattern and Plan Recognition

The problem of *plan recognition* is to induce the plan of action driving an agent's behavior, based on partial observation of its behavior up to the current time. Deriving the underlying plan can be useful for many purposes—predicting the agent's future behavior, interpreting its past behavior, or generating actions designed to influence the plan itself. Researchers in AI have studied plan recognition for several kinds of tasks, including discourse analysis (Grosz and Sidner 1990), collaborative planning (Huber and Durfee 1993), and adversarial planning (Azarewicz et al. 1989). These works have employed a great variety of reasoning techniques,

operating on similarly various plan representations and adopting varied assumptions about observability.

The common theme underlying these diverse motivations and approaches is that the object to be induced is a *plan,* and that this plan is the cause of observed behavior. If there is anything special about the task of plan recognition as opposed to recognition in general, it must be due to special properties of plans: how they are constituted, and how they cause the behavior we observe and wish to predict, interpret, and influence.

We can distinguish plan recognition from uncertain reasoning in general by noting two special features of plans. First, plans are *structured linguistic objects*. Plan languages considered in AI research range from simple sequences of action tokens to general-purpose programming languages. In either case, the recognizer can and should exploit the structure of plans in inducing them from partial observations of the actions comprising the plan. Another way to say this is that plans are descriptions of action *patterns,* and therefore any general pattern-recognition technique is automatically a plan recognition technique for the class of plans corresponding to the class of patterns associated with the given technique.

The second special feature of plans is that they are *rational constructions.* They are synthesized by a rational agent with some beliefs, preferences, and capabilities, that is, a *mental state*. Knowing the agent's mental state and its rationality properties strongly constrains the possible plans it will construct. (The degree of constraint depends on the power of the rationality theory we adopt.) The rational origin of plans is what distinguishes plan recognition from pattern recognition. If the observations available include evidence bearing on the beliefs, preferences, and capabilities of the agent, then the recognizer should combine this with evidence from the observed actions in reasoning about the entire plan.

In prior work (Pynadath and Wellman 1995), we have elucidated a general Bayesian framework for plan recognition. Our basic approach is similar to that of Charniak and Goldman (Charniak and Goldman 1993), elaborating and departing in some respects, less well-developed in others. We describe the high-level idea below; for a more complete description and some specific developments of the technique see the cited papers.

Our framework is *Bayesian* in that we start from a causal theory of how the agent's mental state causes its plan and executing its plan causes activity, and reason from observed effects to underlying causes. Our recognizer has uncertain *a priori* knowledge about the agent's mental state, the world state, and the world's dynamics, which can be summarized (at least in principle) by a probability distribution. It then makes partial observations about the world, and uses this evidence to induce properties of the agent and its plan.

We begin with a model of the planning agent operating in the world. As it begins planning, the agent has a certain mental state, consisting of its preferences (e.g., goals), beliefs (e.g., about the state of its environment), and capabilities (e.g., available actions). We assume the actual planning process to be some rational procedure for generating the plan that will best satisfy the agent's preferences based on its beliefs, subject to its capabilities. This plan then determines (perhaps with some uncertainty) the actions taken by the agent in the world.

Once we have accounted for the agent's plan-generation process, we need to consider the effects of the plan's execution. In many plan-recognition domains, the external observer finds the agent's actions inaccessible. In such cases, the recognizer observes actions only indirectly, via their effects on the world (which themselves are typically only partially observable). These restricted observations then form the basis of inference.

Thus, observations of the state of the world provide two types of evidence about the plan. First, the world influences the agent's initial mental state, which provides the *context* for plan generation. Second, changes in the world state reflect the effects of the agent's actions, which *result* from executing its plan.

To perform plan recognition tasks, we generate a Bayesian network representing the causal planning model and use it to support evidential reasoning from observations to plan hypotheses. The structure of the Bayesian network is based on the framework depicted in Figure 6. That diagram can itself be viewed as a Bayesian network, albeit with rather broad random variables. To make this operational, we replace each component of the model with a subnetwork that captures intermediate structure for the particular problem. The limited connections among the subnetworks reflect the dependency structure of our generic planning model.

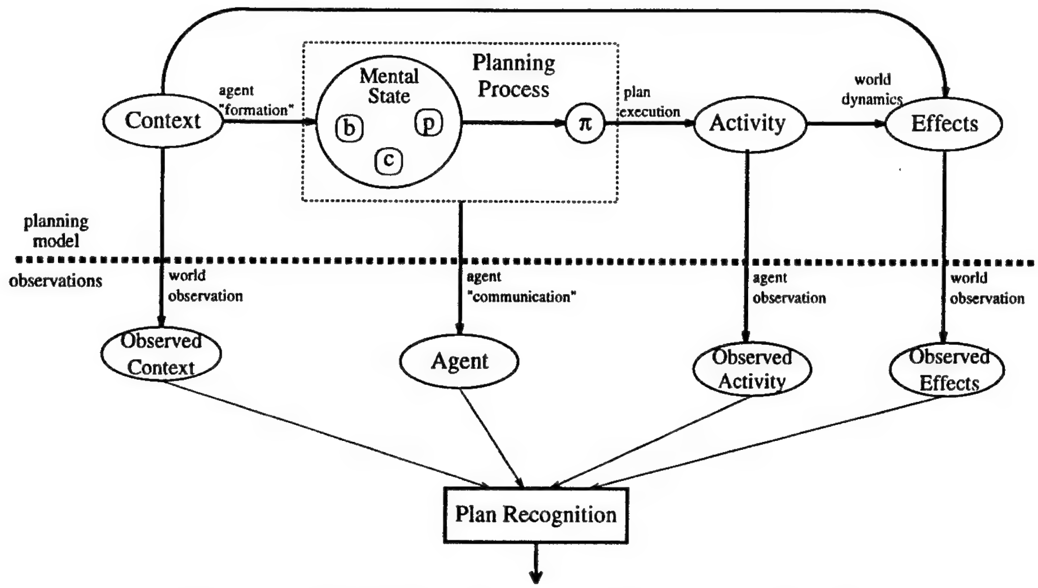

Figure 6: Bayesian plan recognition framework.

The framework as described above is of course very general. We have explored particular instances of the approach, specifically looking at the issue of modeling context in the domain of traffic monitoring (Pynadath and Wellman 1995). In subsequent work, we have begun to investigate more deeply the problem of modeling the planning process. In doing so, we need to adopt particular assumptions about the plans generated, and determine an effective recognition strategy.

Our approach has been to treat plan generation as a structured stochastic process, and recognition as the task of answering queries about events in the generation of particular observations. Our first deep study adopted the generative model of *probabilistic context-free grammars* (PCFGs), a well-studied and commonly applied model for pattern recognition (Wetherell 1980). Interpreting a string of observations generated from a grammar is known as *parsing*, and the general recognition problem can be cast in terms of queries about the parse. For PCFGs, efficient algorithms have been developed for several useful types of queries (i.e., calculating the probability of a given string, or finding the most likely parse). However, for other queries potentially useful in plan recognition, only brute-force enumeration is available.

In our recent work (Pynadath and Wellman 1996), we have extended the class of queries that can be answered in several ways:

(1) allowing missing tokens in a sentence or sentence fragment,

(2) supporting queries about intermediate structure, such as the presence of particular nonterminals, and

(3) flexible conditioning on a variety of types of evidence.

Our method works by constructing a Bayesian network to represent the distribution of parse trees induced by a given PCFG. The network structure mirrors that of the chart in a standard parser, and is generated using a similar dynamic-programming approach. By augmentations of the network, we can relax the context-free restriction of the grammar in a controlled way, admitting important context-sensitivities without invalidating the inferences drawn by the recognizer.

We direct the reader to [that document] for discussion of the technical details of our algorithm. In that document, we present an algorithm for constructing Bayesian networks from PCFGs, and show how queries or patterns of queries on the network correspond to interesting queries on PCFGs.

4.1 Air Force Plan Recognition Application

The generalized pattern-recognition procedure is potentially applicable to a wide range of Air Force problems involving interpreting uncertain or incomplete observations. One example comprises problems of plan recognition, where the aim is to interpret or predict the actions of an observed agent (friend or foe), based on uncertain observations of its action thus far.

One of the more common representations for planning structures used in plan-recognition research is an action decomposition hierarchy, sometimes called an *event tree* (Kautz and Allen 1986). Event trees and other variants of hierarchies map easily to context-free grammars, and indeed the parsing approach to recognition has previously been proposed (for the deterministic case) by Vilain (Vilain 1990). By extending the event-tree model to include probabilities, we provide a basis for distinguishing among equally possible but unequally plausible explanations of the observations. As Charniak and Goldman (Charniak and Goldman 1993) (among others) have argued, this is a critical requirement for any useful plan recognition algorithm.

In air-combat scenarios, for example, we can model the behavior of a fighter plane to allow tracking and prediction of its actions. The probabilistic event tree could include information about possible specializations of its general mission (e.g. fly to target, intercept enemy plane), as well as decompositions of plans into subplans (e.g. employ weapons, evade, chase) or observable actions (e.g. start turning, stop turning, maintain current heading). An example event tree for an air-to-air combat scenario (borrowed from Tambe and Rosenbloom (Tambe and Rosenbloom 1995)) is presented in Figure 7.

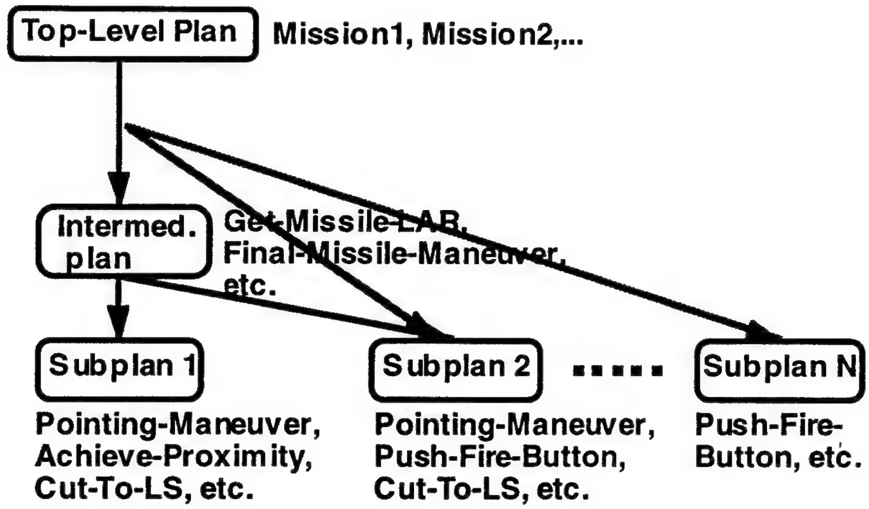

Figure 7: An example event tree from air combat domain.

We can then translate this event tree representation into a probabilistic grammar whose rules correspond to the plan specializations and decompositions. The algorithm mentioned above can use this grammar to generate a Bayesian network corresponding to the probability distribution over the possible behavior of the tracked plane. This network will support a wide variety of useful queries, using the traditional methods of evidence propagation to compute the relevant probabilities. In the air-combat example, a pilot may wish to determine whether a nearby enemy plane is about to launch a missile, or is merely flying to another target. The Bayesian network can provide the probability of either subplan, conditioned on whatever behavior has been observed so far. These probabilities, along with the different implications of the two cases, can aid the pilot in choosing the correct course of action.

5. Market-Based Belief Aggregation

In distributed decision and recognition problems, information, decision authority, and computational resources are decentralized across individual agents. In this project, we have begun to explore the distribution of uncertain reasoning and decision making in a general setting based on economic market mechanisms.

The function of markets as aggregators of uncertain belief are well-recognized. For example, the price of a company's stock represents the "market evaluation" of the expected present value of future dividends, and odds in a horse race aggregate the bettors' beliefs about the winning horse's identity. But despite their commonality and well-developed underlying theory, there has

been little or no work in the uncertain reasoning community on principled application of market ideas for distributed uncertain reasoning.[2]

Our basic approach is to set up markets for uncertain propositions, essentially financial securities that pay off contingent on uncertain events. Agents bid on these propositions according to their beliefs, subject to their wealth constraints and tempered by their confidence and risk aversion. In equilibrium, the market prices can be interpreted as a consensus probability of the participants in the market.

5.1 MarketBayes

To investigate this idea, we performed a study of the expressive power of these kinds of markets compared to standard techniques for probabilistic reasoning. In particular, we constructed an economic model, called *MarketBayes*, that can represent arbitrary joint probability distributions, exploiting dependence structure in a manner similar to Bayesian networks (Pennock and Wellman 1996). Given an arbitrary Bayesian network, our algorithm generates a MarketBayes economy such that the prices of propositions in the unique competitive equilibrium corresponds exactly to probabilities in the Bayesian networks.

Our mapping is depicted schematically in Figure 8. The MarketBayes economy is comprised of two types of agents, consumers and producers, each of which bids on a selected set of uncertain propositions (conjunctions of random variables and their negations). Consumers represent conditional probabilities in terms of the tradeoff between the conditioning proposition and its conjunction with the proposition conditioned. The laws of probability are represented by producers. For example, the producer in Figure 8 arbitrages on the goods A, AB, and AB′, ensuring that the price of the first equals the sum of prices of the latter two.

[2] Though Hanson has presented informed arguments for the idea. In particular, he has advocated setting up a market in scientific claims, where prices summarize consensus opinions about important research questions (Hanson 1995). A prototype of this "Foresight Exchange" is in operation on the World-Wide Web, at `http://www.ideosphere.com/`.

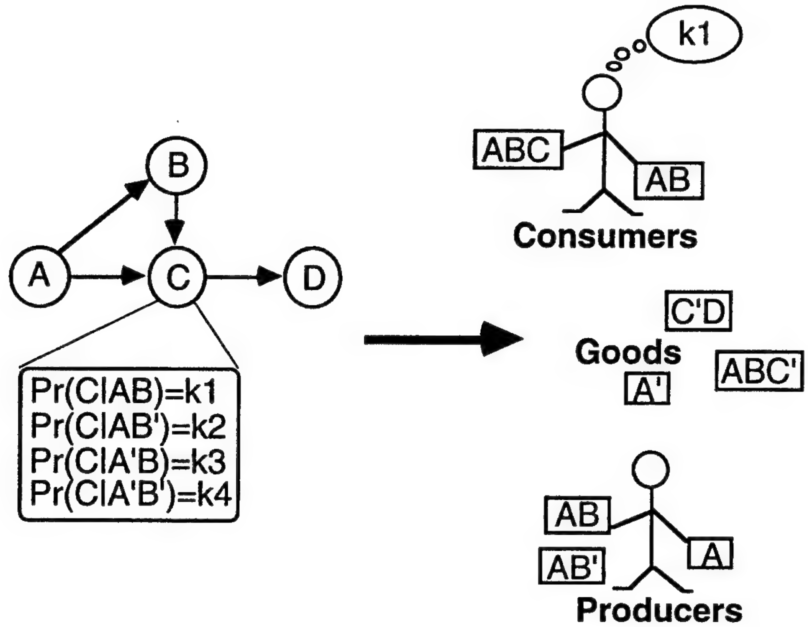

Figure 8: Translation of a Bayesian network to a MarketBayes economy.

MarketBayes has been implemented within our general environment for *market-oriented programming* (Wellman 1993). We have also implemented some securities markets within the Michigan Internet AuctionBot (`http://auction.eecs.umich.edu`), which is our likely platform for further developments.

5.2 Belief Aggregation

The mapping sketched above includes only one agent for each conditional probability in the model. As suggested above, one of the primary motivations for the market approach is belief aggregation, where participating agents may have differing beliefs about the underlying propositions.

To analyze the use of competitive markets as a belief aggregation mechanism, we define *securities* for each event of interest. The security corresponding to event A is worth \$1 if A obtains, and zero otherwise. If the security has an associated price of p, then purchase of x units of the security can be interpreted as a lottery paying off $(1-p)x$ if A occurs, and $-px$ otherwise. Each agent will decide how much of the security to purchase based on the price, its probability for A, and its utility for money.

Starting from the assumptions that agents are competitive, have state-independent risk averse preference for dollars, and do not change belief based on prices, we can characterize the results of running these securities markets

(Pennock and Wellman 1997). Specifically, we have established the following theoretical properties:

1. A risk-averse agent's demand for a single security is positive (zero, negative) if its probability for the corresponding event is greater than (equal to, less than) the security's price.

2. If all agents have constant risk aversion, the equilibrium price is equal to the *normalized logarithmic pool* (or geometric mean, a well known aggregation mechanism (Genest and Zidek 1986)), with individual weights inversely related to the risk aversion coefficients.

3. If the agent believes that a set of events is independent, its decisions about how much to purchase of each security are separable.

4. If two events are dependent, demand for the two are correlated in the opposite direction of the dependence. Intuitively, negatively correlated events provide insurance for each other.

5. Adding a security for an event equivalent to an existing one does not change anything. Similarly for complementary events.

6. If all agents have constant risk aversion, their aggregate behavior is equivalent to a "super agent" with belief equal to the equilibrium price, and risk aversion a function of the individual risk aversion coefficients.

The value of results such as these is to characterize the properties of an aggregation mechanism from the "bottom up". The market-based approach directly addresses self-motivated agents' incentives for participation and truthfulness, and is thus a plausible mechanism for belief aggregation in multiagent systems. We intend to demonstrate the use of this mechanism for distributed decision making problems in future work.

6. Conclusion

In this report, we have documented progress on several problems in uncertain reasoning, most involving some form of automated model construction as a key step. In particular, we have outlined specific results in path planning, abstraction and approximation, pattern and plan recognition, and belief aggregation. Each of these tasks potentially bears on Air Force requirements in various ways, and we are pursuing such applications in ongoing work.

7. References

Azarewicz, J., G. Fala, and C. Heithecker (1989). Template-based multi-agent plan recognition for tactical situation assessment. *Sixth Conference on Artificial Intelligence Applications.*

Charniak, E., and R. P. Goldman (1993). A Bayesian model of plan recognition. *Artificial Intelligence* **64**: 53-79.

Druzdzel, M. J., and M. Henrion (1993). Efficient Reasoning in qualitative probabilistic networks. *Proceedings of the Eleventh National Conference on Artificial Intelligence,* Washington, DC, AAAI Press.

Genest, C., and J. V. Zidek (1986). Combining Probability Distributions: A Critique and an Annotated Bibliography. *Statistical Science* **1**(1): 114-148.

Grosz, B. J., and C. L. Sidner (1990). Plans for discourse. *Intentions in Communication* Ed. P. R. Cohen, J. Morgan, and M. E. Pollack. MIT Press. 417-444.

Hanson, R. D. (1995). Could gambling save science? Encouraging an honest consensus. *Social Epistemology* **9**: 3-33.

Huber, M. H., and E. H. Durfee (1993). Observational uncertainty in plan recognition among interacting robots. *IJCAI Workshop on Dynamically Interacting Robots,* Chambery, France.

Kaufman, D. E., and R. L. Smith (1993). Fastest paths in time-dependent networks for intelligent vehicle-highway systems application. *IVHS Journal* **1**(1).

Kautz, H. A., and J. F. Allen (1986). Generalized plan recognition. *National Conference on Artificial Intelligence,* Philadelphia.

Liu, C.-L., and M. P. Wellman (1995). State-space abstraction for anytime evaluation of Bayesian networks. *IJCAI-95 Workshop on Anytime Algorithms and Deliberation Scheduling,* Montreal.

Liu, C.-L., and M. P. Wellman (1997). Incremental tradeoff resolution in qualitative probabilistic networks. *AAAI-97 Workshop on Abstraction, Decisions, and Uncertainty,* Providence, RI.

Pennock, D. M., and M. P. Wellman (1996). Toward a market model for Bayesian inference. *Twelfth Conference on Uncertainty in Artificial Intelligence,* Portland, OR.

Pennock, D. M., and M. P. Wellman (1997). Representing aggregate belief through the competitive equilibrium of a securities market. *Thirteenth Conference on Uncertainty in Artificial Intelligence*, Providence, RI.

Pynadath, D. V., and M. P. Wellman (1995). Accounting for context in plan recognition, with application to traffic monitoring. *Eleventh Conference on Uncertainty in Artificial Intelligence*, Montréal.

Pynadath, D. V., and M. P. Wellman (1996). Generalized queries on probabilistic context-free grammars. *National Conference on Artificial Intelligence*, Portland, OR, AAAI Press.

Shachter, R. D. (1988). Probabilistic inference and influence diagrams. *Operations Research* **36**: 589-604.

Stewart, B., and C. C. White (1991). Multiobjective A*. *Journal of the Association for Computing Machinery* **38**: 775-814.

Tambe, M., and P. S. Rosenbloom (1995). RESC: An approach for real-time, dynamic agent tracking. *Fourteenth International Joint Conference on Artificial Intelligence*, Montreal.

Vilain, M. (1990). Getting serious about parsing plans. *National Conference on Artificial Intelligence*, Boston.

Wellman, M. P. (1990). Fundamental Concepts of Qualitative Probabilistic Networks. *Artificial Intelligence* **44**: 257-303.

Wellman, M. P. (1993). A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research* **1**: 1-23.

Wellman, M. P., M. Ford, and K. Larson (1995). Path planning under time-dependent uncertainty. *Eleventh Conference on Uncertainty in Artificial Intelligence*, Montréal.

Wellman, M. P., and C.-L. Liu (1994). State-space abstraction for anytime evaluation of probabilistic networks. *Tenth Conference on Uncertainty in Artificial Intelligence*, Seattle, WA.

Wetherell, C. S. (1980). Probabilistic languages: A review and some open questions. *Computing Surveys* **12**(4): 361-379.

Wurman, P. R., and M. P. Wellman (1996). Optimal factory scheduling under uncertainty using stochastic dominance A*. *Twelfth Conference on Uncertainty in Artificial Intelligence*, Portland, OR.

8. Project Information

The following sections describe participants, publications, and interactions arising from this project.

8.1 Personnel Supported

Faculty: Michael P. Wellman

PhD students: MeeSook Hyun, Chao-Lin Liu, David Pynadath

other contributing students (not funded by project): Matthew Ford, Kenneth Larson, David Pennock, Peter Wurman

8.2 Publications

(In chronological order. Many of these are also listed in the alphabetical **References** section of this report.)

M. P. Wellman. Some varieties of qualitative probability. In *Proceedings of the Fifth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 437–442, July 1994. Reprinted in B. Bouchon-Meunier, R. R. Yager, and L. A. Zadeh, editors, *Advances in Intelligent Computing*, Springer Verlag, 1995.

M. P. Wellman and C.-L. Liu. State-space abstraction for anytime evaluation of probabilistic networks. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 567–574, July 1994.

M. J. Huber, E. H. Durfee, and M. P. Wellman. The automated mapping of plans for plan recognition. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 344–351, July 1994.

D. V. Pynadath and M. P. Wellman. Accounting for context in plan recognition, with application to traffic monitoring. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 472–481, August 1995.

M. P. Wellman, M. Ford, and K. Larson. Path planning under time-dependent uncertainty. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 532–539, August 1995.

C.-L. Liu and M. P. Wellman. State-space abstraction for anytime evaluation of Bayesian networks. In *IJCAI-95 Workshop on Anytime Algorithms and Deliberation Scheduling*, August 1995.

M. P. Wellman. The economic approach to artificial intelligence. *ACM Computing Surveys*, 27(3)360–362, 1995.

M. P. Wellman. Rationality in decision machines. *AAAI Fall Symposium on Rational Agency: Concepts, Theories, Models, and Applications*, pages 154–156, November 1995.

D. M. Pennock and M. P. Wellman. Toward a market model for Bayesian inference. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 405–413, August 1996.

D. V. Pynadath and M. P. Wellman. Generalized queries on probabilistic context-free grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1285–1290, August 1996. Revised and extended version submitted for publication.

P. R. Wurman and M. P. Wellman. Optimal factory scheduling using stochastic dominance A*. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 554–563, August 1996.

C.-L. Liu and M. P. Wellman. Incremental tradeoff resolution in qualitative probabilistic networks. In *AAAI-97 Workshop on Abstraction, Decisions, and Uncertainty*, July 1997.

D. M. Pennock and M. P. Wellman. Representing aggregate belief through the competitive equilibrium of a securities market. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, August 1997.

8.3 Interactions

Attended and presented results from this project at the following conferences:

Fifth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, July 1994.
Tenth Conference on Uncertainty in Artificial Intelligence, July 1994.
Eleventh Conference on Uncertainty in Artificial Intelligence, August 1995.
IJCAI-95 Workshop on Anytime Algorithms and Deliberation Scheduling, August 1995
AAAI Fall Symposium on Rational Agency: Concepts, Theories, Models, and Applications, November 1995.
Twelfth Conference on Uncertainty in Artificial Intelligence, August 1996.
Thirteenth National Conference on Artificial Intelligence, August 1996.
AAAI-97 Workshop on Abstraction, Decisions, and Uncertainty, July 1997.
Thirteenth Conference on Uncertainty in Artificial Intelligence, August 1997.

Presented research seminars at:

University of California at Berkeley, University of Wisconsin at Milwaukee, NASA/Ames Research Center, IBM Watson Research Center, Duke University, Kyoto University, NTT Communication Science Laboratories, Brown University, Carnegie Mellon University, Stanford University, Siemens Corporate Research, Pennsylvania State University